DOCUMENTATION FOR PDF ASSISTANT

AI PDF Assistant

A web-based application that allows users to upload PDF documents and interact with them through natural language questions. The application uses Google's Generative AI to provide accurate responses based on the PDF content.

Features

- PDF document upload and processing
- Real-time question-answering capabilities
- Interactive chat interface
- Document context maintenance
- Support for multiple document types
- WebSocket-based communication for real-time responses

Prerequisites

Before running the application, ensure you have the following installed:

- Python 3.8 or higher
- pip (Python package installer)
- Google API key for Generative AI

Installation

1. Clone the repository:

bash

git clone <repository-url>

cd AIplanet

2. Create a virtual environment:

bash

python -m venv venv

3. Activate the virtual environment:

- Windows:
bash

venv\Scripts\activate
bash

source venv/bin/activate

4. Install required packages:

bash

pip install -r requirements.txt

Required Packages

Create a `requirements.txt` file with the following dependencies: fastapi

uvicorn

python-multipart

python-dotenv

langchain

langchain-google-genai

google-generativeai

PyPDF2

faiss-cpu

sqlalchemy

Environment Setup

1. Create a `.env` file in the root directory

2. Add your Google API key:

Running the Application

1. Start the server:
bash

uvicorn app:app –reload
2. Open your web browser and navigate to:

http://localhost:8000

Usage

1. **Upload PDF**
   - Click "Choose PDF" button
   - Select your PDF file
   - Click "Upload PDF" button
   - Wait for confirmation message

2. **Ask Questions**
   - Type your question in the input field
   - Press Enter or click "Ask" button
   - Wait for the AI response

Features in Detail

PDF Processing

- Extracts text from PDF documents
- Creates embeddings for efficient searching
- Stores document context for future reference

Question Answering
- Uses Google's Generative AI
- Maintains conversation context
- Provides relevant answers based on document content

User Interface
- Clean, modern design
- Real-time response display
- Error handling and status updates
- PDF file name display
- Chat-like interface for Q&A

Error Handling

The application includes error handling for:
- Invalid file types
- Upload failures
- Processing errors
- API connection issues
- Invalid questions

Security Features

- Rate limiting on API endpoints
- File type validation

- Maximum file size restrictions
- Secure WebSocket connections

Limitations

- Currently supports PDF files only
- Maximum file size limit
- Requires active internet connection
- API key rate limits apply

Troubleshooting

1. **PDF Upload Fails**
   - Check file size
   - Ensure PDF is not corrupted
   - Verify file permissions

2. **No Response to Questions**
   - Check internet connection
   - Verify API key is valid

1 Ensure PDF was uploaded successfully

2 Server Won't Start

3 check if port 8000 is available

4 Verify all dependencies are installed

5 Ensure Python version is compatible

Contributing

1 Fork the repository

2 Create your feature branch

3 Commit your changes

4 Push to the branch

5 Create a new Pull Request

License

This project is licensed under the MIT License - see the LICENSE file for details

Acknowledgments

1 Google Generative AI

2 FastAPI framework

3 LangChain library

4 FAISS by Facebook Research

Support

For support, please open an issue in the repository or contact the maintainers.

Future Enhancements

1.Support for more document types

2.Multi-language support

3.Document comparison features

4.Enhanced error handling

5.User authentication

6.Document history tracking

7.Export conversation feature

This README provides:

1. Clear installation instructions

2. Usage guidelines

3. Project structure

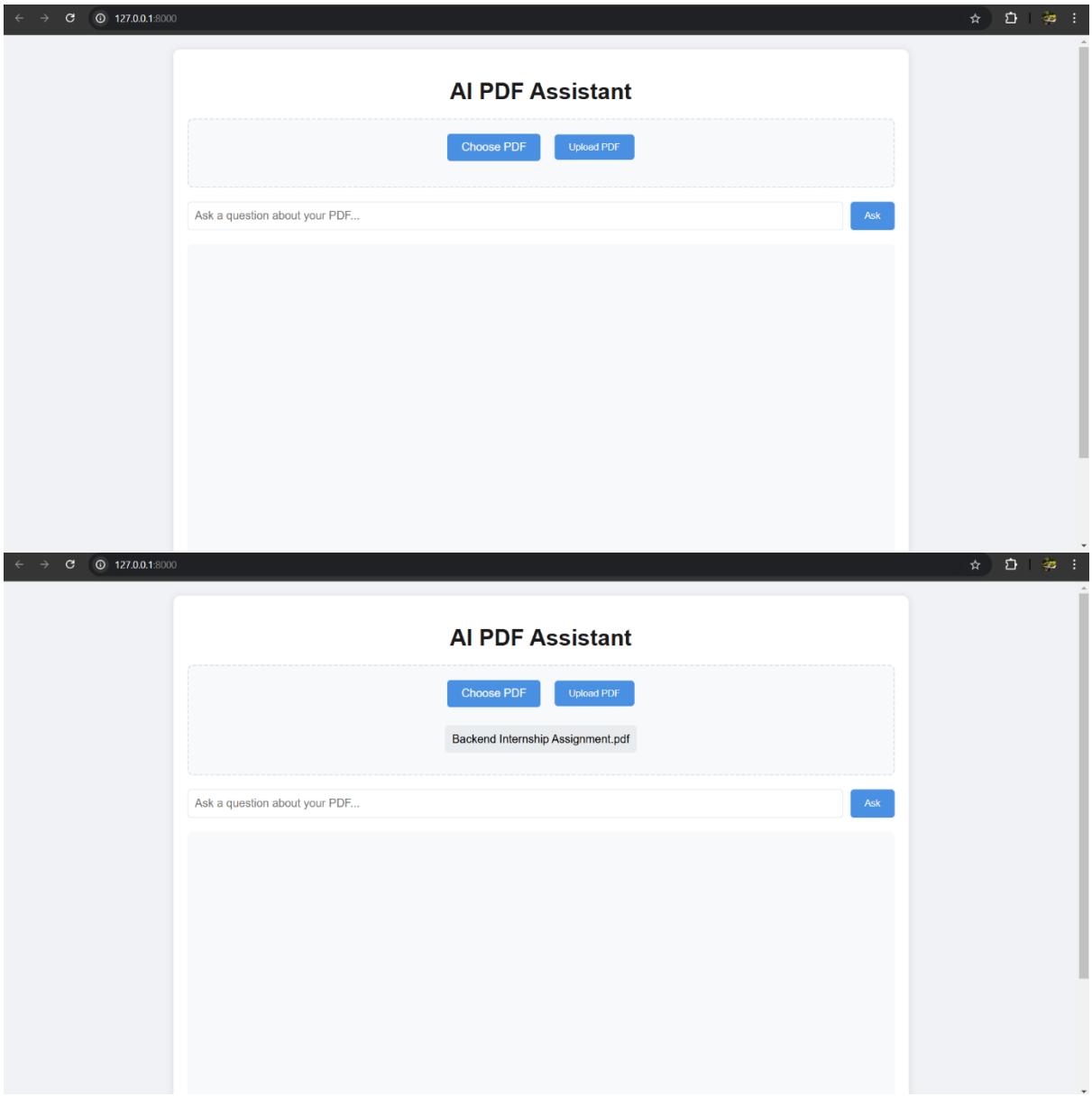
4. Troubleshooting tips

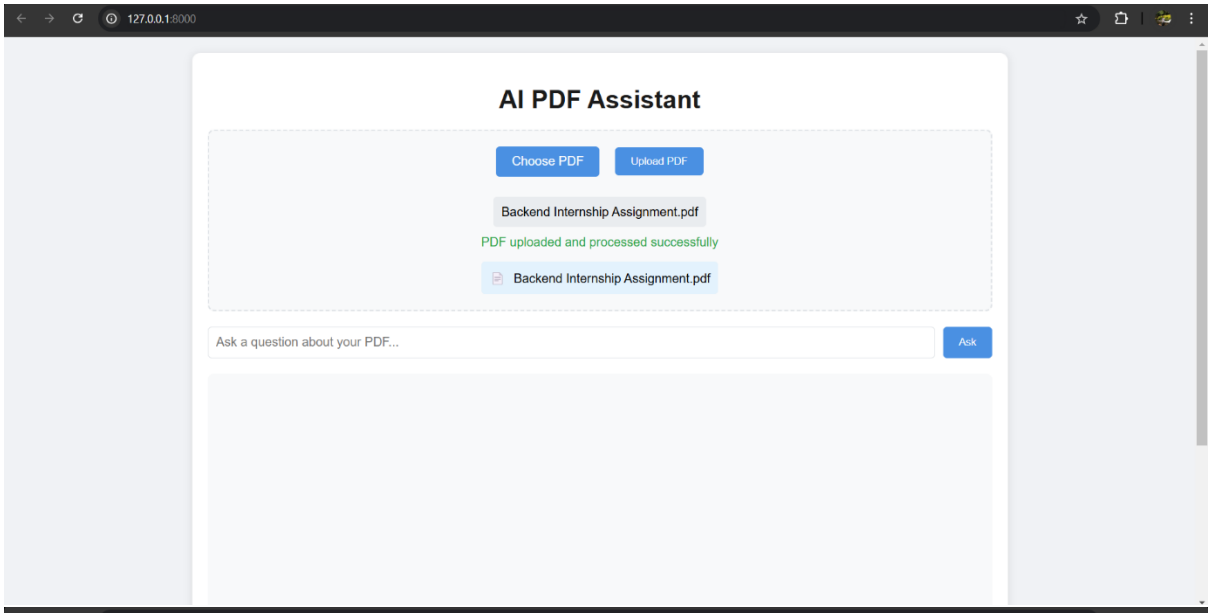
5. Future enhancement plans

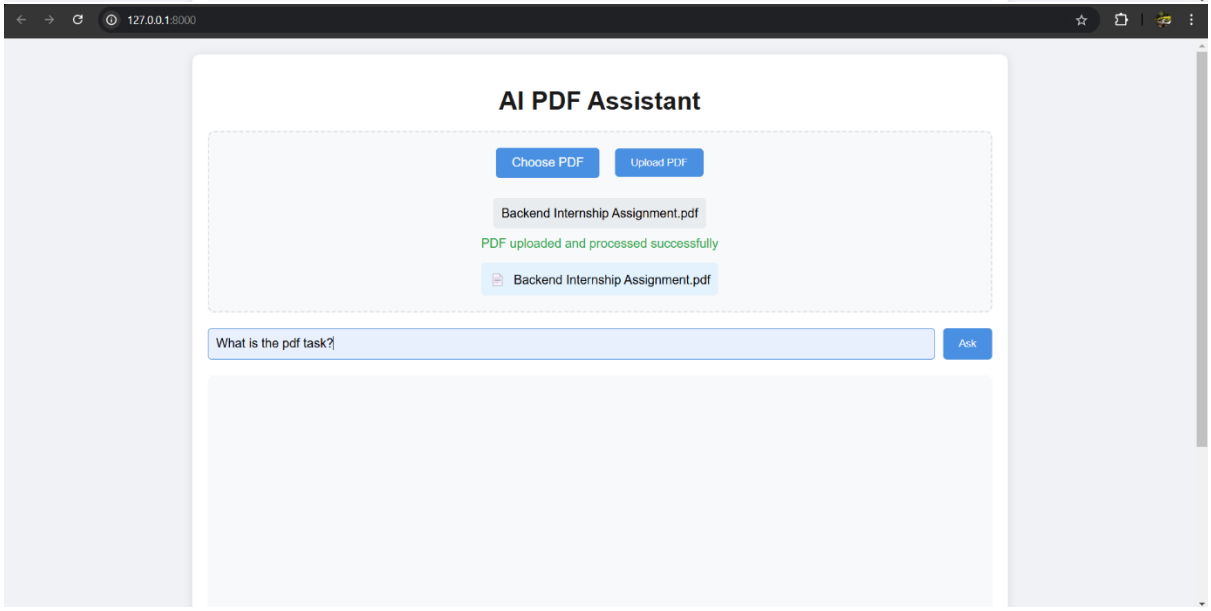
6. Security considerations

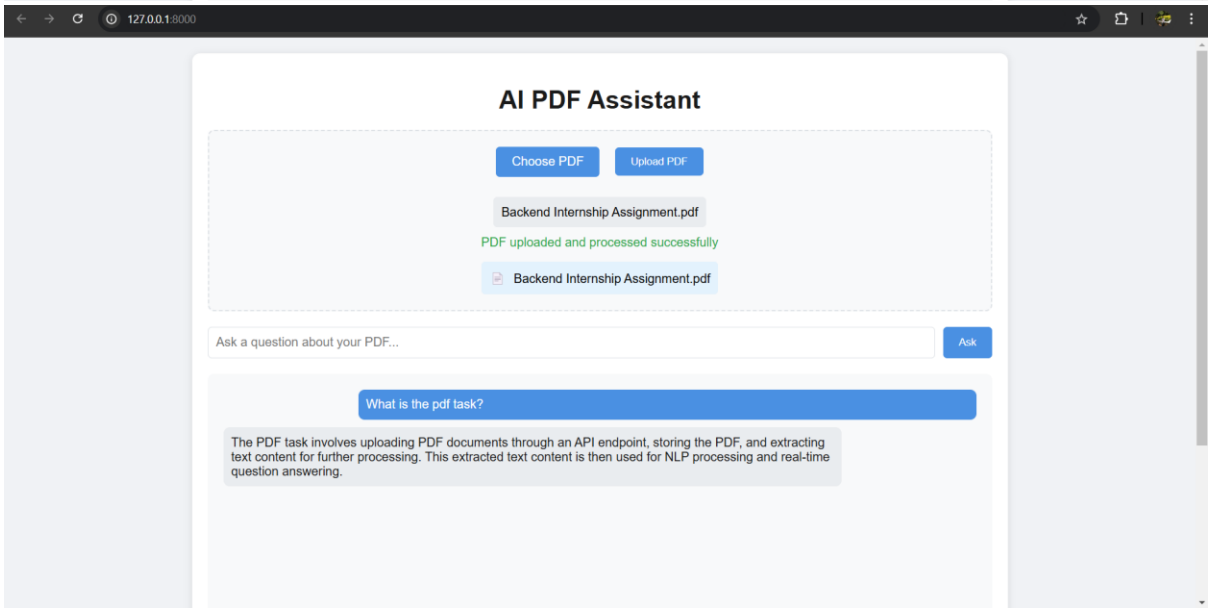
7. Detailed feature explanations

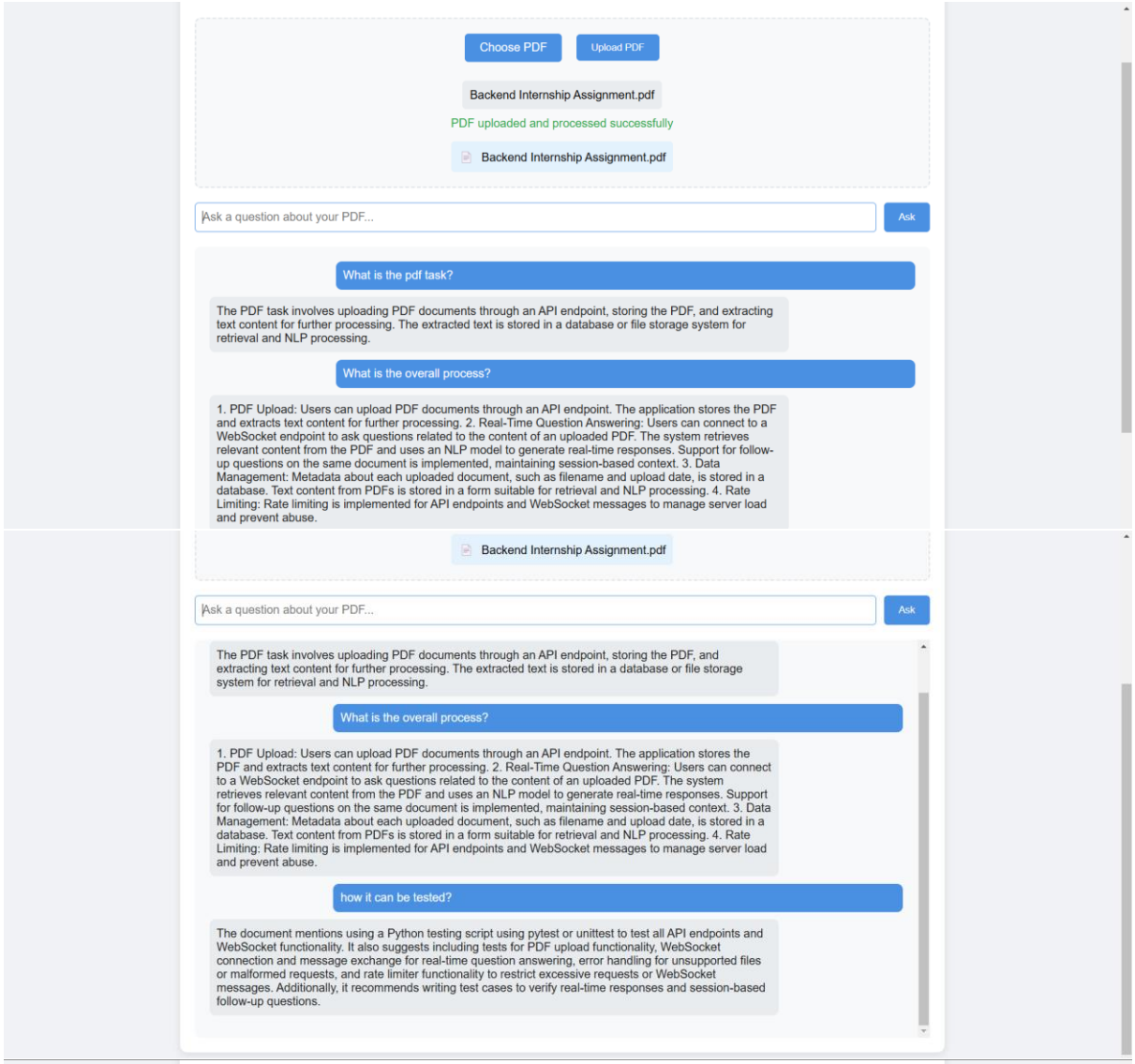
Output :

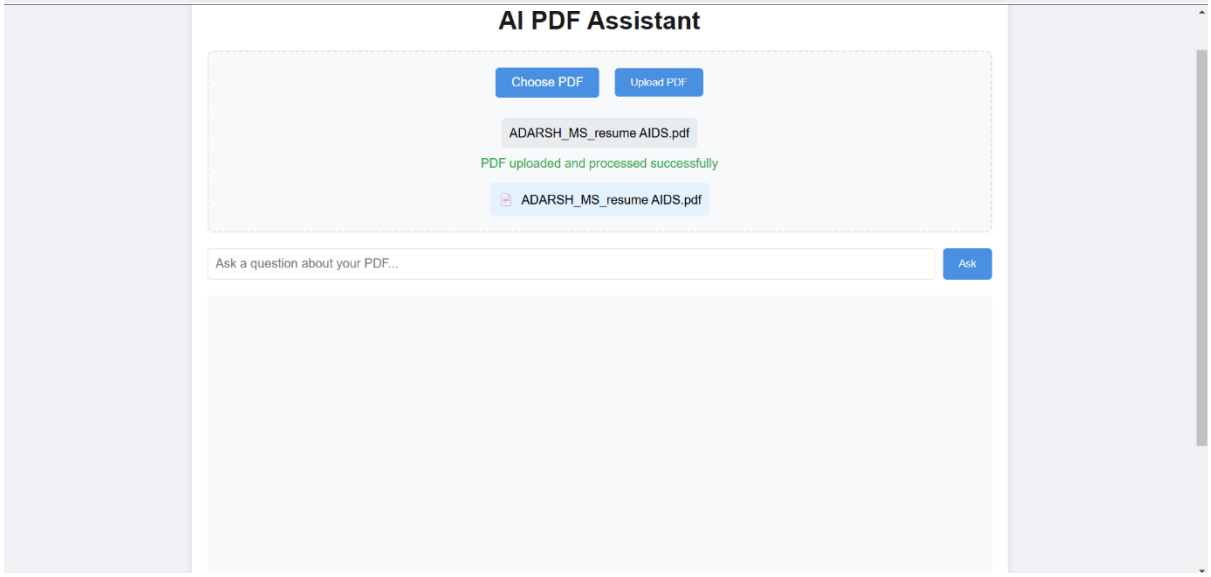

127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
Ask a question about your PDF...
Ask
127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
Backend Internship Assignment.pdf
Ask a question about your PDF...
Ask

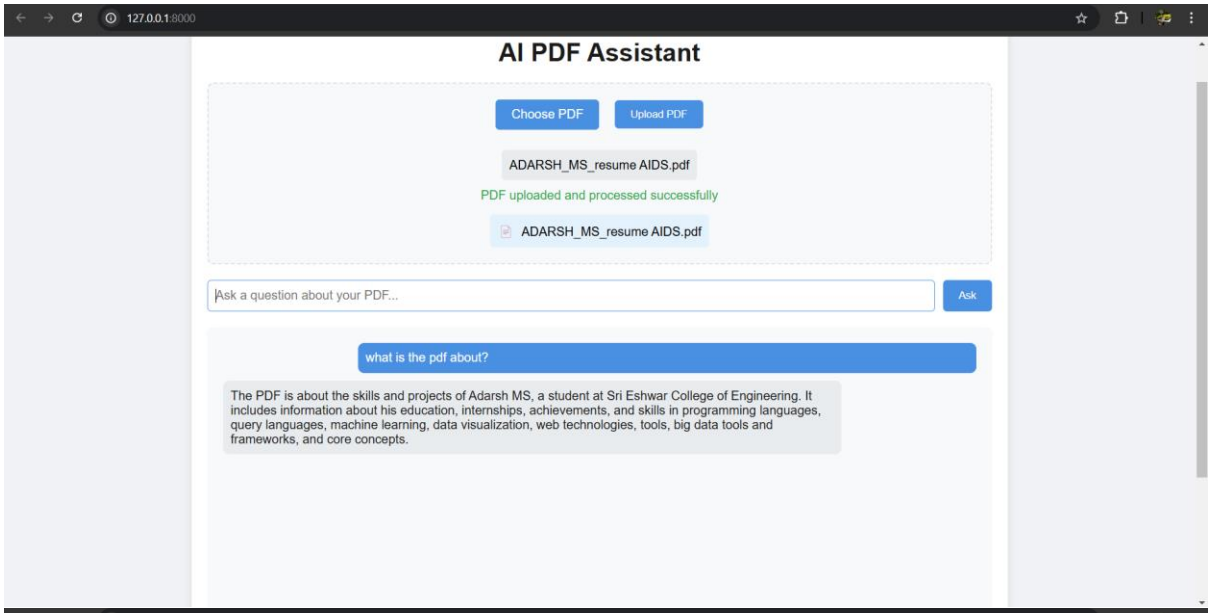

127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
Backend Internship Assignment.pdf
PDF uploaded and processed successfully
Backend Internship Assignment.pdf
Ask a question about your PDF...
Ask

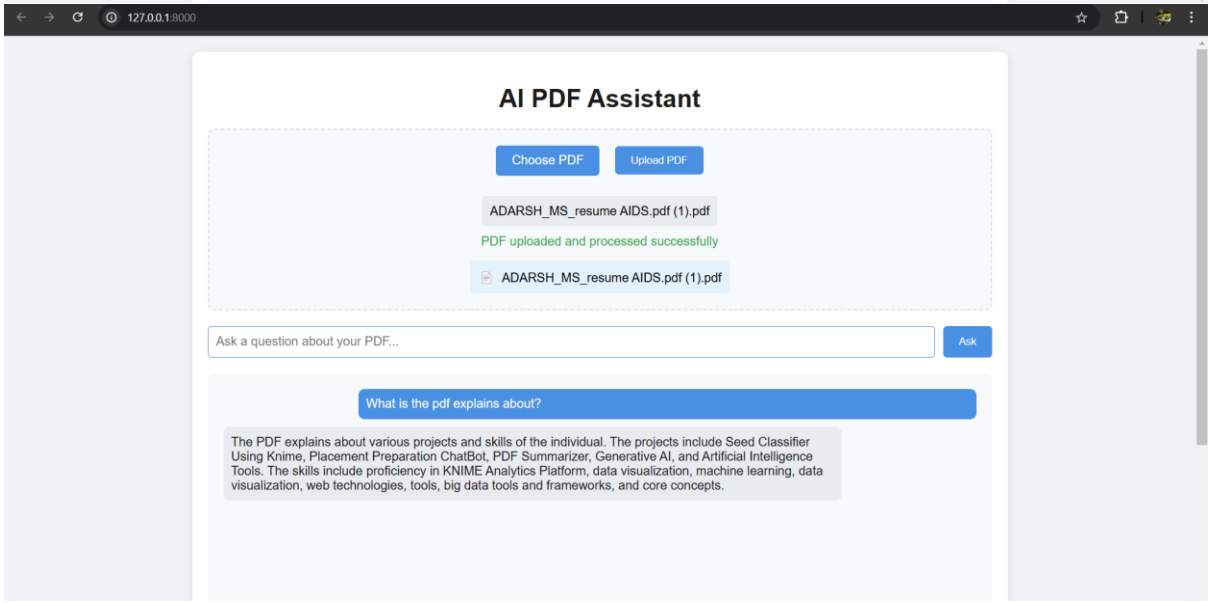

127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
Backend Internship Assignment.pdf
PDF uploaded and processed successfully
Backend Internship Assignment.pdf
What is the pdf task?
Ask

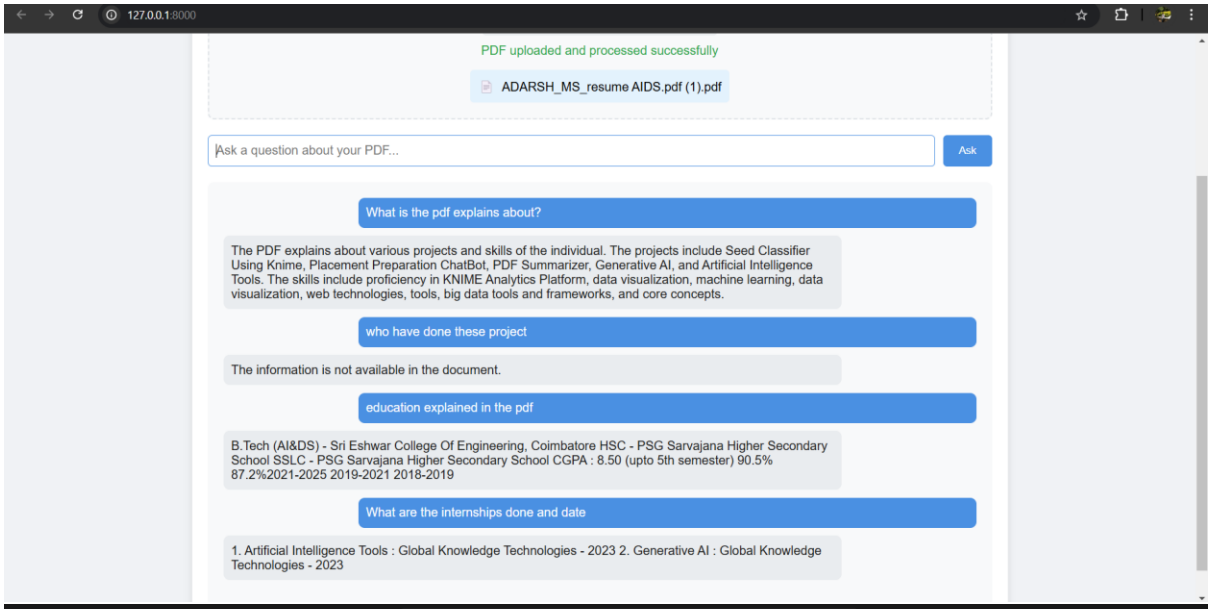

127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
Backend Internship Assignment.pdf
PDF uploaded and processed successfully
Backend Internship Assignment.pdf
Ask a question about your PDF...
Ask
What is the pdf task?
The PDF task involves uploading PDF documents through an API endpoint, storing the PDF, and extracting text content for further processing. This extracted text content is then used for NLP processing and real-time question answering.

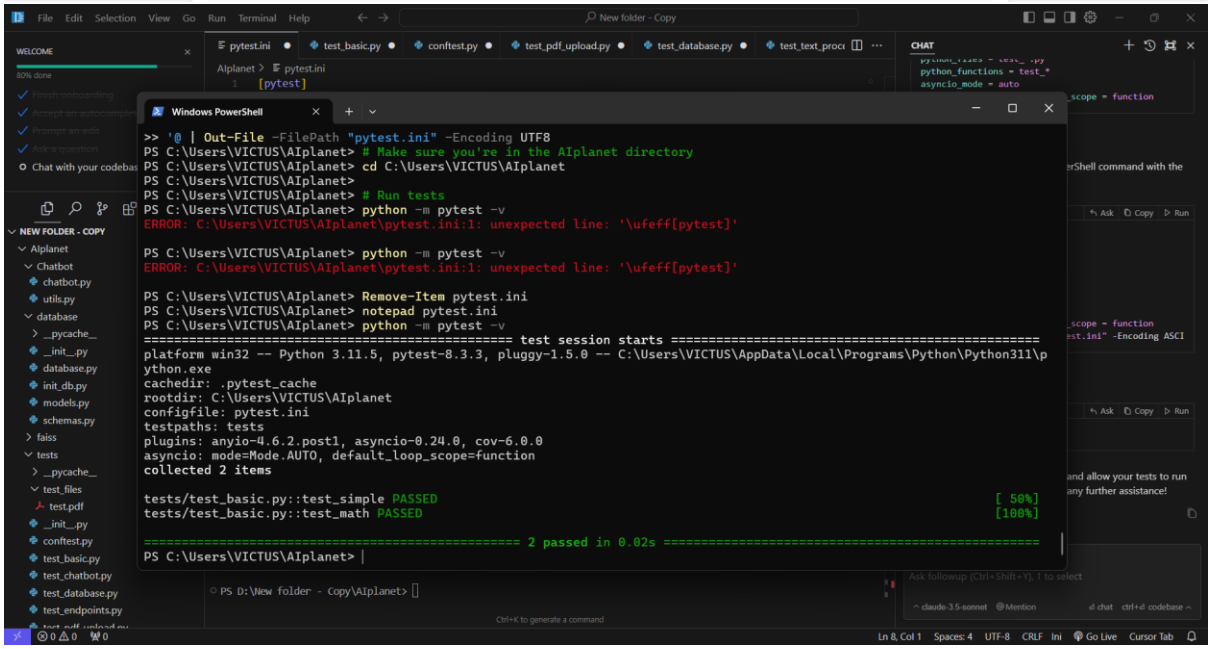

Choose PDF
Upload PDF
Backend Internship Assignment.pdf
PDF uploaded and processed successfully
Backend Internship Assignment.pdf
Ask a question about your PDF...
Ask
What is the pdf task?
The PDF task involves uploading PDF documents through an API endpoint, storing the PDF, and extracting text content for further processing. The extracted text is stored in a database or file storage system for retrieval and NLP processing.
What is the overall process?
1. PDF Upload: Users can upload PDF documents through an API endpoint. The application stores the PDF and extracts text content for further processing. 2. Real-Time Question Answering: Users can connect to a WebSocket endpoint to ask questions related to the content of an uploaded PDF. The system retrieves relevant content from the PDF and uses an NLP model to generate real-time responses. Support for follow-up questions on the same document is implemented, maintaining session-based context. 3. Data Management: Metadata about each uploaded document, such as filename and upload date, is stored in a database. Text content from PDFs is stored in a form suitable for retrieval and NLP processing. 4. Rate Limiting: Rate limiting is implemented for API endpoints and WebSocket messages to manage server load and prevent abuse.
Backend Internship Assignment.pdf
Ask a question about your PDF...
Ask
The PDF task involves uploading PDF documents through an API endpoint, storing the PDF, and extracting text content for further processing. The extracted text is stored in a database or file storage system for retrieval and NLP processing.
What is the overall process?
1. PDF Upload: Users can upload PDF documents through an API endpoint. The application stores the PDF and extracts text content for further processing. 2. Real-Time Question Answering: Users can connect to a WebSocket endpoint to ask questions related to the content of an uploaded PDF. The system retrieves relevant content from the PDF and uses an NLP model to generate real-time responses. Support for follow-up questions on the same document is implemented, maintaining session-based context. 3. Data Management: Metadata about each uploaded document, such as filename and upload date, is stored in a database. Text content from PDFs is stored in a form suitable for retrieval and NLP processing. 4. Rate Limiting: Rate limiting is implemented for API endpoints and WebSocket messages to manage server load and prevent abuse.
how it can be tested?
The document mentions using a Python testing script using pytest or unittest to test all API endpoints and WebSocket functionality. It also suggests including tests for PDF upload functionality, WebSocket connection and message exchange for real-time question answering, error handling for unsupported files or malformed requests, and rate limiter functionality to restrict excessive requests or WebSocket messages. Additionally, it recommends writing test cases to verify real-time responses and session-based follow-up questions.


AI PDF Assistant
Choose PDF
Upload PDF
ADARSH_MS_resume AIDS.pdf
PDF uploaded and processed successfully
ADARSH_MS_resume AIDS.pdf
Ask a question about your PDF...
Ask


127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
ADARSH_MS_resume AIDS.pdf
PDF uploaded and processed successfully
ADARSH_MS_resume AIDS.pdf
Ask a question about your PDF...
Ask
what is the pdf about?
The PDF is about the skills and projects of Adarsh MS, a student at Sri Eshwar College of Engineering. It includes information about his education, internships, achievements, and skills in programming languages, query languages, machine learning, data visualization, web technologies, tools, big data tools and frameworks, and core concepts.


127.0.0.1:8000
AI PDF Assistant
Choose PDF
Upload PDF
ADARSH_MS_resume AIDS.pdf (1).pdf
PDF uploaded and processed successfully
ADARSH_MS_resume AIDS.pdf (1).pdf
Ask a question about your PDF...
Ask
What is the pdf explains about?
The PDF explains about various projects and skills of the individual. The projects include Seed Classifier Using Knime, Placement Preparation ChatBot, PDF Summarizer, Generative AI, and Artificial Intelligence Tools. The skills include proficiency in KNIME Analytics Platform, data visualization, machine learning, data visualization, web technologies, tools, big data tools and frameworks, and core concepts.


127.0.0.1:8000
PDF uploaded and processed successfully
ADARSH_MS_resume AIDS.pdf (1).pdf
Ask a question about your PDF...
Ask
What is the pdf explains about?
The PDF explains about various projects and skills of the individual. The projects include Seed Classifier Using Knime, Placement Preparation ChatBot, PDF Summarizer, Generative AI, and Artificial Intelligence Tools. The skills include proficiency in KNIME Analytics Platform, data visualization, machine learning, data visualization, web technologies, tools, big data tools and frameworks, and core concepts.
who have done these project
The information is not available in the document.
education explained in the pdf
B.Tech (AI&DS) - Sri Eshwar College Of Engineering, Coimbatore HSC - PSG Sarvajana Higher Secondary School SSLC - PSG Sarvajana Higher Secondary School CGPA : 8.50 (upto 5th semester) 90.5% 87.2%2021-2025 2019-2021 2018-2019
What are the internships done and date
1. Artificial Intelligence Tools : Global Knowledge Technologies - 2023 2. Generative AI : Global Knowledge Technologies - 2023


Windows PowerShell
>> '@ | Out-File -FilePath "pytest.ini" -Encoding UTF8
PS C:\Users\VICTUS\AIplanet> # Make sure you're in the AIplanet directory
PS C:\Users\VICTUS\AIplanet> cd C:\Users\VICTUS\AIplanet
PS C:\Users\VICTUS\AIplanet>
PS C:\Users\VICTUS\AIplanet> # Run tests
PS C:\Users\VICTUS\AIplanet> python -m pytest -v
ERROR: C:\Users\VICTUS\AIplanet\pytest.ini:1: unexpected line: '\ufeff[pytest]'
PS C:\Users\VICTUS\AIplanet> python -m pytest -v
ERROR: C:\Users\VICTUS\AIplanet\pytest.ini:1: unexpected line: '\ufeff[pytest]'
PS C:\Users\VICTUS\AIplanet> Remove-Item pytest.ini
PS C:\Users\VICTUS\AIplanet> notepad pytest.ini
PS C:\Users\VICTUS\AIplanet> python -m pytest -v
============================= test session starts =============================
platform win32 -- Python 3.11.5, pytest-8.3.3, pluggy-1.5.0 -- C:\Users\VICTUS\AppData\Local\Programs\Python\Python311\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\VICTUS\AIplanet
configfile: pytest.ini
testpaths: tests
plugins: anyio-4.6.2.post1, asyncio-0.24.0, cov-6.0.0
asyncio: mode=Mode.AUTO, default_loop_scope=function
collected 2 items
tests/test_basic.py::test_simple PASSED [50%]
tests/test_basic.py::test_math PASSED [100%]
============================== 2 passed in 0.02s ==============================
PS C:\Users\VICTUS\AIplanet>
NEW FOLDER - COPY
Aiplanet
Chatbot
chatbot.py
utils.py
database
database.py
init_db.py
models.py
schemas.py
faiss
tests
test_files
test.pdf
conftest.py
test_basic.py
test_chatbot.py
test_database.py
test_endpoints.py
CHAT